Towards Statistical Machine Translation with Unification Grammars

Exam Number: 9877703

Master of Science

Speech and Language Processing

School of Philosophy, Psychology and Language Sciences

University of Edinburgh

2009

Abstract

Traditional Statistical Machine Translation (SMT) models account poorly for many linguistic phenomena, such as subject-verb agreement and differences in word-order between languages. Recent work, such as that in factored phrase-based models, has shown promising improvements in translation quality through the use of linguistically-richer models. Unification-based approaches to grammar offer a framework for modelling agreement, a particular problem in generating morphologically-rich languages, and so in order to gauge the potential gains available from their application to SMT we first consider how to automatically recognise and measure agreement failure. We focus upon the specific issue of declension in German noun phrases and propose a simple unification-based approach to the problem. We develop an agreement checker based on this approach and use it to assess the agreement failure rate of a hierachical phrase-based translation system trained on the small News Commentary corpus. Initially we find that our checker reports unreasonably high failure rates on the fluent training data, and through an incremental process of failure analysis and lexicon refinement we significantly reduce the number of spurious failures. We then apply the agreement checker directly to machine translation by incorporating it as a feature function of the log-linear model. We train our baseline system on the larger Europarl corpus and again measure failure rates before applying the agreement check as both a hard and soft constraint. The effects on translation are not large enough to reliably measure using standard automatic evaluation techniques and so we perform a manual analysis of the types of change introduced.

Acknowledgements

I am indebted to my supervisor, Philipp Koehn, for his support throughout this project. It was his suggestion to explore the application of unification-based approaches to statistical machine translation, and his guidance that nudged me away from wild unfeasibility and towards shaping a manageable, rewarding, and not least, fun, masters project. Vielen thanks!

I would also like to thank Hieu Hoang, Adam Lopez, and Miles Osborne for their valuable suggestions and help.

To Gillian, thanks for making this busy few months less stressful and so much more enjoyable.

Declaration

I have read and understood The University of Edinburgh guidelines on Plagiarism and declare that this written dissertation is all my own work except where I indicate otherwise by proper use of quotes and references.

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Overview 2

1.3 Structure of the Dissertation 2

2 Background **4**

2.1 Statistical Machine Translation 4

2.2 Morphology in German 7

2.3 Unification Grammars 9

3 Detecting Agreement Failure in Machine Translation **14**

3.1 Overview 14

3.2 Complexity in Language 14

3.3 Automatically Recognising Agreement Failure 15

3.4 Summary 20

4 Developing the Agreement Checker **21**

4.1 Overview 21

4.2 Learning the Initial Grammar Rules 21

4.3 Learning the Initial Lexicon 23

4.4 Implementation 24

4.5 Results for the Training Data 26

4.6 Results for Translation Output 31

4.7 Summary 37

5 A Feature Function for Agreement **38**

5.1 Overview 38

5.2 The Baseline Translation Model 38

5.3 Agreement Results 39

5.4 Developing and Integrating the Feature Function 40

5.5 Agreement as a Hard Constraint 40
5.6 Agreement as a Soft Constraint 44

6 Conclusions and Further Work **45**
6.1 Conclusions 45
6.2 Directions for Future Work 45

Bibliography **48**

Chapter 1

Introduction

1.1 Motivation

Since the introduction of IBM's word-based models in the early 1990s, empirical approaches have come to dominate machine translation research. By exploiting the implicit language knowledge captured by parallel corpora and monolingual n-gram models, a statistical machine translation (SMT) system can produce (perhaps surprisingly) useful translations, and given an SMT toolkit and appropriate data, an implementer with no personal knowledge of a language pair can rapidly build such a system.

Whilst the use of increasingly large training sets continues to improve performance (see Brants et al (2007) for an extreme example of n-gram language modelling), traditional models still account poorly for many linguistic phenomena, such as subject-verb agreement or differences in word-order between language pairs, and much recent work has shown promising improvements through the use of linguistically-richer models. For instance, in factored phrase-based models (as explored in the 2006 JHU Language Engineering Workshop (Koehn et al, 2007b)) words are represented as vectors of linguistic components — such as part-of-speech, lemma, and morphological features — and independent translations of the individual components may all inform the generation of the final output forms.

For syntax, word-order differences in many language pairs can be better accounted for using hierarchical phrase-based models (Chiang, 2005), using explicit hand-crafted reordering rules (Collins et al, 2005), or with a separate statistical framework (Cowan et al, 2006).

Particularly challenging in machine translation is the generation of morphologically-rich languages, for which the larger vocabulary exacerbates both data sparsity and agreement issues. Approaches so far include factored translation, as just mentioned, and the prediction of morphology using information from both source and target sides (Minkov et al, 2007).

In computational linguistics, the need to look beyond surface form representations and to formally encode linguistic features and constraints is addressed by the class of constraint-

based, or unification, grammars. These associate words and syntactic constituents with feature structures: objects that capture linguistic attributes. Constraints are expressed through identities that must hold in order for a rule to apply and that confer features from smaller to larger constituents through an operation known as unification.

Unification grammars have been proposed for machine translation previously, (for example, Kay (1984) describes a theoretical model based on Functional Unification Grammar), but we are unaware of any application to statistical machine translation so far.

1.2 Overview

In this dissertation we focus upon one specific — yet ubiquitous — issue of German morphology: that of declension within noun phrases. As a means of gauging the potential benefit of a unification-based approach to SMT, we first consider how agreement failures in the output of a machine translation system might automatically be recognised and therefore measured.

We propose a simple unification-based approach to this problem whereby the set of possible declensional interpretations for a phrase is searched for a consistent interpretation. If none can be found then we consider the phrase not to agree.

We develop an agreement checker based on this approach and use it to assess the agreement failure rate of a hierachical phrase-based translation system trained on the small News Commentary corpus. The agreement checker's lexicon is extracted from a parse of the German-side of the corpus and (with a couple of fairly major qualifications) is therefore adequate for any translation produced by the same system.

Initially we find that our checker reports high failure rates on the fluent training data, and through an incremental process of failure analysis and lexicon refinement we significantly reduce the number of spurious failures. After testing on translation output, two further refinements are proposed and implemented with the aim of increasing the strength of failure detection.

Finally we apply the agreement checker directly to machine translation by incorporating it as a feature function in the now-standard log-linear model. We train our baseline system on the larger Europarl corpus and again measure failure rates. We then implement the feature function and apply the agreement check as both a hard and soft constraint. The effects on translation are not large enough to reliably measure using standard automatic evaluation techniques and so we perform a manual analysis of the types of change introduced.

1.3 Structure of the Dissertation

The organisation of the remaining text is as follows:

Chapter 2 provides very brief introductions to the three major topics of this dissertation: statistical machine translation, German morphology, and unification-based approaches to grammar.

Chapter 3 proposes and describes our unification-based approach to testing agreement within German noun phrases.

Chapter 4 describes the initial implementation, testing, and development of the approach described in chapter 3. Most of the work is focussed on the refinement of our extracted lexicon.

Chapter 5 describes the development of a feature function based on the method developed so far. The agreement check is applied as both a hard and soft constraint and results are presented for a hierarchical phrase-based system trained on the Europarl corpus. We provide an analysis of the types of change introduced between the baseline system and the test system.

Chapter 6 presents our conclusions and suggests directions for future work.

Chapter 2

Background

This chapter provides very brief introductions to the three main topics of this dissertation: statistical machine translation, German morphology, and unification grammars. Each has a wide literature and we provide a few pointers to core texts.

2.1 Statistical Machine Translation

The field of statistical machine translation emerged in the early 1990s lead by the work of Brown et al (1990) on word-based models. With the development of statistical techniques for automatically aligning parallel corpora — bilingual or multilingual text sources — at the sentence and word levels, and the application of the noisy channel model, an idea from information theory that had earlier been applied to automatic speech recognition, SMT systems were built that could compete with their then-dominant rule-based counterparts.

The basic principles of these systems were carried over into early phrase-based models, of which variants now represent the state of the art, and so they deserve a brief description here.

2.1.1 The Noisy Channel Model

At the core of the noisy channel SMT model are two components: a bilingual translation model and a monolingual (target-side) language model. They are employed by a decoder in the task of finding the most probable sequence of target-language words or phrases given the source sentence. In other words, the best translation according to the probabilistic model.

A decoder is thus trying to find the optimal translation, t, given the source, s, which according to Bayes' theorem is given by[1]

$$\arg\max_t P(t|s) = \arg\max_t P(s|t)P(t)$$

$P(s|t)$ and $P(t)$ are modelled by the translation and language models, respectively.

[1]The denominator $P(s)$ has been dropped since the source sentence is constant during any given search.

In a word-based system, the translation model tells us the conditional probability of a source word given a target word. For example, the probability $P(hund|dog)$ if our source language is German and our target language is English. In a phrase-based system, a 'phrase' is an arbitrary sequence of one or more words and our translation model tells us the conditional probability of a source phrase given a target phrase.

The language model tells us the probability of a target word conditioned on its history. In practice, as in other fields of statistical natural language processing, this is usually a low-order n-gram model. So it might tells us $P(dog|the, big)$, and hopefully that it's greater than $P(big|the, dog)$.

This model is intended to capture two ideals of translation: adequacy and fluency. If our source text is discussing a *Hund*, a translation should probably mention a 'dog'. And we'll probably have an easier time reading about 'a big dog' than about 'a dog big'.

2.1.2 The Log Linear Model

Motivated by the desire to incorporate additional model components and to apply scaling factors to both the original and new components, Och and Ney (2002) reformulated the problem as a log linear model:

$$\arg\max_t P(t|s) = \arg\max_t \sum_{i=1}^{n} \lambda_i h_i(t,s)$$

Here, there are an arbitrary number, n, of model components, each implemented as a feature function, $h_i(t,s)$, with a corresponding scaling factor, λ_i. The translation and language models of the noisy channel model are typically used as two such feature functions.

In Och et al (2004), the authors investigate the addition of a broad range of feature functions, including a word-based translation model (their baseline model is phrase-based), a n-gram model over part-of-speech tags, and a probabilistic parser. Most of the features are found to offer small incremental improvements, the most significant being the word-based translation model.

Moses (Koehn et al, 2007a), a state-of-the-art phrase-based SMT system, typically uses the language model and phrase translation components of the noisy channel model, together with an inverse phrase translation model (that is, one estimating $P(t|s)$); word-based models in both directions; a configurable reordering model, which provides a basic model of word-order differences between the source and target languages; and two constant values that may be weighted to reward or penalise lexical production and phrase length.

2.1.3 Parameter Estimation

The true probability distributions of the model's components are of course unknown, and must be estimated from data. In practice, the translation models are estimated from parallel cor-

pora, including multilingual news sources and sources such as Europarl (Koehn, 2005), an 11-language sentence-aligned corpus extracted from over a decade's worth of European parliamentary proceedings.

IBM's original series of word-based models (Brown et al, 1993) were trained using expectation maximisation algorithms that search for the most probable alignments between the words of the corpus's sentence pairs (and are guaranteed to at least find local maxima). Tools for performing this alignment, and predominantly Och's GIZA++, usually still form the basis of model estimation in phrase-based models since phrasal-alignments are inferred from word-alignments.

Language models are usually n-gram based, as in speech recognition and other fields of statistical natural language processing. Monolingual corpora are inherently more abundant than parallel corpora, and n-gram language models are now routinely trained from hundreds of millions of words.

2.1.4 Hierarchical Phrase-Based Models

The hierarchical phrase-based model (Chiang, 2005) is an extension of the phrase-based model in which grammar rules are derived from phrase pairs through the replacement of subphrases with a single non-terminal. For example, from the English-German phrase pair,

(*the big brown dog*, *der große braune Hund*)

are extracted grammar rules such as,

$$X \rightarrow \langle \textit{the X dog}, \textit{der X Hund} \rangle$$

and

$$X \rightarrow \langle \textit{the X}, \textit{der X} \rangle$$

Chiang (2005) implements such a system based around a chart-parser decoding algorithm and presents results for the Mandarin-English language pair showing that this approach can significantly improve translation quality.

2.1.5 Further Reading

A comprehensive introduction to the field up to and including the state-of-the-art is given in Koehn (forthcoming). Introductions to the principles of early SMT are given in Knight (1997) and Manning and Schütze (1999).

2.2 Morphology in German

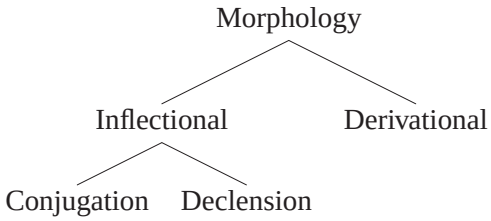
In both English and German, morphology can be divided into two branches: inflectional and derivational. Inflectional morphology deals with the formation, from a root morpheme, of surface-form words within a single word class. For example, the formation of 'plays,' the 3rd-person singular present tense verb, and 'played,' the simple past tense verb, from the root 'play.' Or similarly in German, the formation of *spielt* and *spielte* from *spiel*.

In the case of verbs, this process is called conjugation and although a richer set of categorical distinctions is made, is similar in German to English.

A separate process, *declension*, applies to all German determiners, attributive adjectives, nouns, and pronouns. In English, declension is limited mainly to the formation of plural noun forms and to distinctions of clausal role, such as between 'he' and 'him,' within the pronouns.

The second branch, derivational morphology, deals with the formation of forms belonging to different word classes. For example, the formation of the noun 'player' and the adjective 'playable' from the verb 'play.'

This broad categorisation of German morphology is therefore as follows:



In this dissertation, we are concerned only with inflectional morphology, since that is where we find the issue of agreement. In the following sections, we give a brief introduction to declension in German.

2.2.1 Declension

As an illustration of the function of declension in German, consider the following (somewhat repetitive) English sentence and a German translation:

The big man	throws	the big dog	the big stick	.
Der große Mann	*wirft*	*dem großen Hund*	*den großen Stock*	*zu*.

Der, *dem*, and *den* all mean 'the,' but are declined to indicate grammatical case as well as gender and number. Similarly, *große* and *großen* both mean 'big,' and are declined according to the same three grammatical categories.

Both sentences contain a subject, a finite verb, an indirect object, and a direct object, in that order (the final German word, *zu*, is a separable prefix and forms part of the verb). Whereas in English this order is fairly fixed — we can't say 'the big dog throws the big man the big stick'

without changing the meaning — in German the roles of the clause elements are indicated by grammatical case and so the order is less important. We can therefore write, for example,

Dem großen Hund wirft der große Mann den großen Stock zu.

without introducing any confusion about who's throwing what. Rather, the choice of the second form over the first would emphasises the fact that the recipient of the stick is the dog.

2.2.2 Case, Number, and Gender

German has four grammatical cases. Of these, the nominative, accusative, and dative are primarily used to indicate the roles of the clause elements in relation to the main verb: nominative is used for the subject, accusative for the direct object, and dative for the indirect object. The fourth case, the genitive, is used primarily to indicate possession. For example,

Der Stock	*des Hund.*
The stick	of the dog.

Like in English, there is a distinction between two classes of grammatical number: singular and plural. Most nouns have distinct forms for each: for instance, singular *Hund* and plural *Hunde*. There are seven regular plural endings (compared to English, which with few exceptions uses the suffix '-s'), for which there is a limited degree of predictability.

Additionally, every noun belongs to one[2] of three gender classes: masculine, feminine or neuter. The genders of approximately 80% of nouns can be predicted from the noun's surface form or meaning (Durrell, 2002), though the rules are numerous and often highly specific.[3]

2.2.3 Agreement

Verbs are conjugated to indicate number, person, tense, mood, and voice. We do not describe verb conjugation or the latter four categories here, but instead focus on the declension of determiners, adjectives, and nouns.

Determiners and attributive adjectives are declined to agree with the number and gender of their corresponding nouns as well as the case in which they occur. Nouns are declined to show number and in limited contexts also mark case.

Generally, the declension of a word in isolation does not unambiguously indicate the grammatical classes to which it belongs. For example, the adjective *großen* is both a masculine, accusative, singular form and a masculine, dative, singular form. In fact there are only five adjective suffixes: *-e*, *-em*, *-en*, *-er*, and *-es*.

[2] A small number of nouns have varying genders, often indicating distinct meanings. For instance, *der See* ('lake') and *das See* ('sea').

[3] For example: alcoholic and plant-based drinks are masculine (with *das Bier* being an exception); aeroplanes, motorbikes, and ships are feminine (Durrell, 2002).

Similarly there are only six surface form words for the definite article: *das*, *dem*, *den*, *der*, *des*, and *die*, despite there being 16 combinations of case, gender, and number (gender is never reflected in plural forms, otherwise that number would be 24).

However the combination of endings in a phrase, such as *dem großen Hund* or *den großen Stock*, together indicate the grammatical classes of the constituent words, usually without ambiguity.

In addition to case, gender, and number, the choice of adjective ending is further affected by the type of determiner used, or the absence thereof. Where no determiner is used, the ending of the adjective must carry more grammatical information, and is taken from the full set of five suffixes. This is referred to as *strong* declension. Where the definite article is used, only two adjective endings are used between all 16 case, gender, and number combinations. This is referred to as *weak* declension. For other determiners, a combination of the two declensions, sometimes referred to as *mixed* declension, is used.

2.2.4 Further Reading

Necessarily, this section has presented a highly simplified view of only a few aspects of German grammar. There are numerous English-language books on the subject, but Johnson (1997) is noteworthy for providing a linguistically-oriented introduction, and Durrell (2002) is a thorough and practical reference.

2.3 Unification Grammars

Traditional phrase structure grammars allow us to define syntactic rules for a language and to define unequivocally which sentences belong to the language and which do not.

Suppose that we want to describe a language containing the two sentences, ‘the crocodile swims’ and ‘the elephants trumpet’ (probably among others, though it’s a perfectly good language). A natural, English-like set of phrase structure grammar rules might express the ideas that a determiner and a noun together form a larger sentence constituent: the noun phrase; that a noun phrase and a verb phrase form a complete sentence; and that a verb phrase need just be a verb.

Expressed as rewrite rules, a minimal grammar expressing these ideas is

S	→	NP VP
NP	→	Det Noun
VP	→	Verb

And when combined with a lexicon that associates words with our lexical categories:

Det	→	‘the’	Verb	→	‘swims’
Noun	→	‘crocodile’	Verb	→	‘trumpet’
Noun	→	‘elephants’			

it allows the two sentence derivations as desired:

S	→	NP VP	S	→	NP VP
S	→	NP Verb	S	→	NP Verb
S	→	NP ‘swims’	S	→	NP ‘trumpet’
S	→	Det Noun ‘swims’	S	→	Det Noun ‘trumpet’
S	→	Det ‘crocodile’ ‘swims’	S	→	Det ‘elephants’ ‘trumpet’
S	→	‘the’ ‘crocodile’ ‘swims’	S	→	‘the’ ‘elephants’ ‘trumpet’

However, it also allows us to derive ‘the elephants swims’ and ‘the crocodile trumpet’, which of course in English are ungrammatical (or at least nonsensical, if we misinterpret ‘trumpet’ as a noun). If our intention is to develop a grammar that replicates the agreement rules of English then we might remedy this by introducing more restrictive syntactic categories:

S	→	NP-sg-3rd VP-sg-3rd	VP-sg-3rd	→	V-sg-3rd
S	→	NP-pl-3rd VP-pl-3rd	VP-pl-3rd	→	V-pl-3rd
NP-sg-3rd	→	Det Noun-sg-3rd			
NP-pl-3rd	→	Det Noun-pl-3rd			

and a stricter lexicon:

Det	→	‘the’	V-sg-3rd	→	‘swims’
Noun-sg-3rd	→	‘crocodile’	V-pl-3rd	→	‘trumpet’
Noun-pl-3rd	→	‘elephants’			

But this bloats our grammar, potentially a serious problem if parsing performance is affected by grammar size (which in practice is likely to be unavoidable since we have to store and search the rule set, though note that parsing complexity is $O(n^3)$ on sentence length for the chart parsing, binarised-CKY, and Earley algorithms (Grune and Jacobs, 2008)).

Naturally, this is a bigger problem in a language, such as German, that exhibits a richer level of agreement at the surface-form level.

Unification-based formalisms offer an alternative approach. The linguistic attributes of words or constituents, such as grammatical number, are instead contained within distinct objects known as feature structures. Rather than encoding grammatical constraints in the non-terminals, they are expressed as identities that must hold in order for a rule to apply.

There are a number of such formalisms — Functional Unification Grammar (FUG), Generalized Phrase Structure Grammar (GPSG), and Head-Driven Phrase Structure Grammar (HPSG), among them. Here we follow the terminology and notation of Shieber (1986) and describe the simpler PATR-II formalism.

2.3.1 Feature Structures

A feature structure maps names to values. Values may either be atomic symbols, as is *sg* in

$$\begin{bmatrix} \text{NUMBER} & sg \end{bmatrix}$$

or they may themselves be feature structures[4], as is the value of the AGREEMENT feature in

$$\begin{bmatrix} \text{CATEGORY} & \textit{noun} \\ \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix}$$

A nested feature can be referenced by specifying its *path*, the sequence of intervening features from outer- to innermost. So in the feature structure above, the path ⟨AGREEMENT NUMBER⟩ refers to the feature with value *sg*.

Values may be indexed and shared between features such that a change to one value also changes the others. This is called reentrancy and is an important concept of the PATR-II formalism. Since we don't use it in the basic feature structures required for the work of this dissertation, we omit a further description.

2.3.2 Subsumption and Unification

One feature structure is said to *subsume* another if it contains (only) a subset of the information contained in the other. For example, the feature structure

$$\begin{bmatrix} \text{CATEGORY} & \textit{np} \end{bmatrix} \qquad (D_1)$$

subsumes both itself and the more specific

$$\begin{bmatrix} \text{CATEGORY} & \textit{np} \\ \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix} \qquad (D_2)$$

But it does not subsume either

$$\begin{bmatrix} \text{CATEGORY} & \textit{vp} \\ \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix} \qquad (D_3)$$

or

$$\begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix} \qquad (D_4)$$

Of the latter two, D_3 is actually inconsistent with D_1 (since they contain differing NUMBER values), whereas D_4 is not. The respective information of D_1 and D_4 is able to coexist within a single feature structure and the smallest such feature structure is said to be their *unification*. The unification of D_1 and D_4 is in fact D_2.

The symbol $\sqsubseteq$ is used to denote the subsumption relation and the symbol $\sqcup$ is used to denote the unification operator. So we can write $D_1 \sqsubseteq D_2$ and $D_1 \sqcup D_4 = D_2$.

[4]Technically, in PATR-II a value is always a feature structure. An atomic symbol, like *sg*, is a *simple* feature structure whereas a feature-value pairing is a *complex* feature structure. Here we use the term 'feature structure' to refer to complex feature structures only.

2.3.3 Constraints

A constraint on rule application, such as that the number of a verb agrees with that of its subject, is expressed through one or more identities relating the features of the relevant constituents.

Our problematic S rule from earlier can be rewritten as

$$\begin{aligned} &\text{S} \rightarrow \text{NP VP} \\ &\quad \langle \text{NP AGREEMENT} \rangle = \langle \text{VP AGREEMENT} \rangle \end{aligned}$$

If we apply a bottom-up interpretation to rule application then often a rule will confer features from a constituent on the right hand side to the resulting larger constituent. In the following grammar rule, the feature structure of the resulting NP 'inherits' the AGREEMENT value from the feature structure of the Noun:

$$\begin{aligned} &\text{NP} \rightarrow \text{Det Noun} \\ &\quad \langle \text{NP AGREEMENT} \rangle = \langle \text{Noun AGREEMENT} \rangle \end{aligned}$$

2.3.4 The Grammar

The grammar is a set of context-free grammar rules with associated identities, as just described, together with a lexicon: a mapping from surface-form strings to feature structures.

Our example grammar can now be rewritten as a PATR-II unification-based grammar. The grammar rules are

$$\begin{aligned} &\text{S} \rightarrow \text{NP VP} \\ &\quad \langle \text{NP AGREEMENT} \rangle = \langle \text{VP AGREEMENT} \rangle \end{aligned}$$

$$\begin{aligned} &\text{NP} \rightarrow \text{Det Noun} \\ &\quad \langle \text{NP AGREEMENT} \rangle = \langle \text{Noun AGREEMENT} \rangle \end{aligned}$$

$$\begin{aligned} &\text{VP} \rightarrow \text{Verb} \\ &\quad \langle \text{VP AGREEMENT} \rangle = \langle \text{Verb AGREEMENT} \rangle \end{aligned}$$

and the lexicon is

$$
\begin{aligned}
\textit{the} \longmapsto\ & \begin{bmatrix} \textsc{category} & \textit{det} \end{bmatrix} \\
\textit{elephants} \longmapsto\ & \begin{bmatrix} \textsc{category} & \textit{noun} \\ \textsc{agreement} & \begin{bmatrix} \textsc{number} & \textit{pl} \\ \textsc{person} & \textit{3rd} \end{bmatrix} \end{bmatrix} \\
\textit{crocodile} \longmapsto\ & \begin{bmatrix} \textsc{category} & \textit{noun} \\ \textsc{agreement} & \begin{bmatrix} \textsc{number} & \textit{sg} \\ \textsc{person} & \textit{3rd} \end{bmatrix} \end{bmatrix} \\
\textit{swims} \longmapsto\ & \begin{bmatrix} \textsc{category} & \textit{verb} \\ \textsc{agreement} & \begin{bmatrix} \textsc{number} & \textit{sg} \\ \textsc{person} & \textit{3rd} \end{bmatrix} \end{bmatrix} \\
\textit{trumpet} \longmapsto\ & \begin{bmatrix} \textsc{category} & \textit{verb} \\ \textsc{agreement} & \begin{bmatrix} \textsc{number} & \textit{pl} \\ \textsc{person} & \textit{3rd} \end{bmatrix} \end{bmatrix}
\end{aligned}
$$

2.3.5 Further Reading

Shieber (1986) and Jurafsky and Martin (2008, chapter 15) both give clear introductions to unification-based approaches. In addition to describing the PATR-II formalism, Shieber (1986) presents an overview of the major concepts of several richer unification-based grammar formalisms.

Chapter 3

Detecting Agreement Failure in Machine Translation

3.1 Overview

In investigating the application of unification-based approaches to SMT, our primary motivation is the desire to produce translations that are more grammatical and, in particular, that exhibit better morphological agreement. In order to gauge the potential benefit of a such an approach, we would first like to measure the rate of agreement failure in a state-of-the-art SMT system. In this chapter we focus on the specific issue of intra-noun phrase agreement in German and propose a simple method for recognising agreement failure.

3.2 Complexity in Language

Like English, German allows the construction of complex noun phrases through the use of pre- and postmodifiers, such as adjectival phrases, relative clauses, and prepostitional phrases. The following example, taken from Europarl, contains three nouns, and much longer examples certainly aren't uncommon:

> . . . *eine Angelegenheit, die am Donnerstag zur Sprach kommen wird*[1] . . .

Fortunately, the issue of deciding which words should agree does not generally require a comprehensive syntactic analysis. Though an adjectival phrase might come between a determiner and its noun, and might itself contain a noun phrase, in most cases a determiner or adjective will agree with the first noun that follows. Of course, language use is complicated and it's rare that we can state a simple rule without immediately running into exceptions. But for now, let's state the following heuristic:

[1] Roughly, 'an issue that will come up on Thursday'

Heuristic 1 *A determiner or attributive adjective will usually agree with the first noun that succeeds it.*

And hope we can dodge most of the sticky syntactic issues.

3.3 Automatically Recognising Agreement Failure

Suppose we want to write a program capable of checking the agreement of isolated single-noun noun phrases, such as *der große Hund* or *ein Krokodil*. For now, we'll ignore the problem of extracting noun phrases from their surrounding text (though heuristic 1 suggests it might not be too tricky).

There are already freely-available language-processing tools that can provide sophisticated morphological analyses of German text,[2] so in principle we could write a simple wrapper that presents the noun phrase to the language tool and then checks the interpretation that it receives back. However, our objective is to check the output from a SMT system, which is often very badly-formed. It is unclear how well-suited a tool will be if it is trained on a treebank or otherwise expects to receive fluent German.

Instead, let's consider how we'd test agreement given only the surface form words. An obvious unification-based approach would be to implement a recogniser for a grammar that admits phrases if they agree and rejects them otherwise.

We would like its lexicon to contain entries for the words we are testing:

$$
\begin{aligned}
\textit{der} \longmapsto & \begin{bmatrix} \text{POS} & \textit{art} \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & \textit{nom} \\ \text{GENDER} & \textit{masc} \\ \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix} \\
\textit{große} \longmapsto & \begin{bmatrix} \text{POS} & \textit{adja} \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & \textit{nom} \\ \text{GENDER} & \textit{masc} \\ \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix} \\
\textit{Hund} \longmapsto & \begin{bmatrix} \text{POS} & \textit{nn} \\ \text{AGREEMENT} & \begin{bmatrix} \text{GENDER} & \textit{masc} \\ \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix}
\end{aligned}
$$

and we would like it to know appropriate grammar rules, such as

$$
\begin{aligned}
&\text{NP} \rightarrow \text{ART ADJA NN} \\
&\quad \langle \text{ART AGREEMENT GENDER} \rangle = \langle \text{NN AGREEMENT GENDER} \rangle \\
&\quad \langle \text{ART AGREEMENT NUMBER} \rangle = \langle \text{NN AGREEMENT NUMBER} \rangle \\
&\quad \langle \text{ART AGREEMENT} \rangle = \langle \text{ADJA AGREEMENT} \rangle
\end{aligned}
$$

[2] Such as BitPar: `http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/BitPar.html`

A challenge with taking this approach is that an adequate lexicon must include conflicting agreement interpretations for many individual surface-form words: as we saw in section 2.2, a word's surface form, in isolation, doesn't uniquely indicate its grammatical role. For example, in addition to the interpretation above, *der* is also the genitive, plural form of the definite article:

$$der \longmapsto \begin{bmatrix} \text{POS} & art \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & gen \\ \text{NUMBER} & pl \end{bmatrix} \end{bmatrix}$$

as well as the dative, feminine, singular form of the relative pronoun:

$$der \longmapsto \begin{bmatrix} \text{POS} & prels \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & dat \\ \text{GENDER} & fem \\ \text{NUMBER} & sg \end{bmatrix} \end{bmatrix}$$

along with several other interpretations.

For recognising agreement failure, we needn't actually decide on a particular morphological interpretation of any word, we need only check the existence or absence of a consistent interpretation. For example, it is unclear how to interpret

* *der grüne Krokodil*

but it is certainly incorrect because *Krokodil* is a neuter noun and there is no neuter interpretation of *der*.[3]

Now suppose that we are considering the valid noun phrase,

die schwarze Katze

that we have the grammar rule from earlier,

NP → ART ADJA NN
⟨ART AGREEMENT GENDER⟩ = ⟨NN AGREEMENT GENDER⟩
⟨ART AGREEMENT NUMBER⟩ = ⟨NN AGREEMENT NUMBER⟩
⟨ART AGREEMENT⟩ = ⟨ADJA AGREEMENT⟩

and that our lexicon has reasonable entries for *die*, *schwarze*, and *Katze*. Among their respective feature structure sets should be some with part-of-speech values that match the right-hand side of the rule. We are then looking for a combination of agreement feature interpretations that is consistent with the identities of the rule. This situation is illustrated in figure 3.1, which shows the full set of agreement interpretations for *die*, *schwarze*, and *katze*, from left to right, together with the two consistent sequences.

[3]Note that we're assuming *der grüne Krokodil* is a noun phrase, so *der* must be an article. Even if *der* is interpreted as a relative pronoun (in which case it will agree with a preceeding noun), then the adjective *grüne* should still agree with the noun *Krokodil*, which it does not.

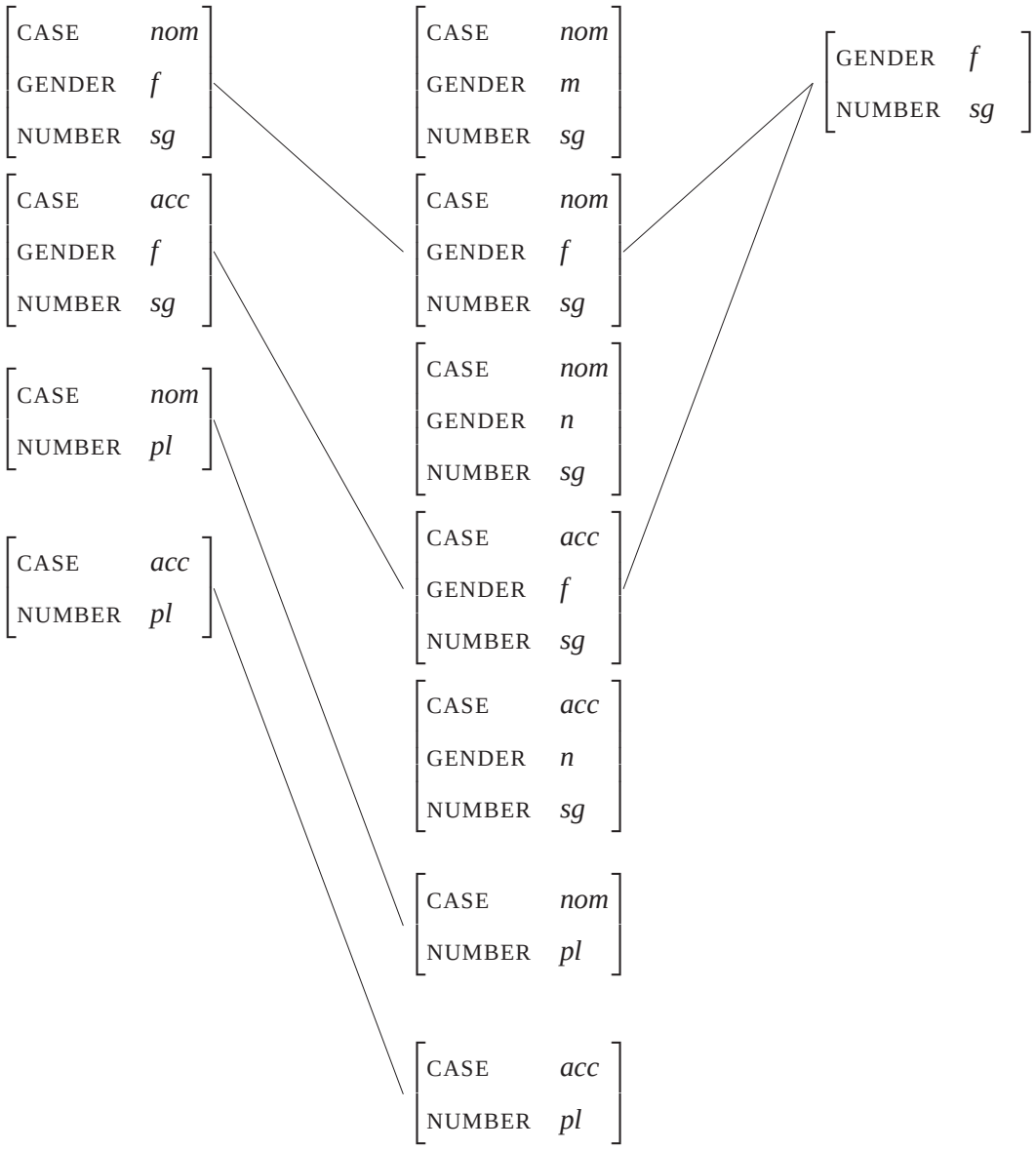


Figure 3.1: Consistent interpretations of the noun phrase *die schwarze Katze* (and partial sequences *die schwarze* and *schwarze Katze*) under the rule NP → ART ADJA NN.

In fact, for the purposes of checking agreement, our rules' identities will all express the same basic constraint: that the agreement features of the relevant words are compatible. The feature structures won't necessarily be identical, since some will contain more information than others, but they should be unifiable.

If we abuse the notation slightly then we can rewrite the earlier rule as something like

$$\begin{aligned}&\text{NP} \rightarrow \text{ART ADJA NN}\\ &\quad \exists D_1 : D_1 = \langle \text{ART AGREEMENT} \rangle \sqcup \langle \text{ADJA AGREEMENT} \rangle\\ &\quad \exists D_2 : D_2 = \langle \text{ADJA AGREEMENT} \rangle \sqcup \langle \text{NN AGREEMENT} \rangle\end{aligned}$$

In other words, the grammar rule applies if the agreement feature structure of our article can be unified with that of our adjective, and if the agreement feature structure of our adjective can be unified with that of our noun.

If we can define a set of NP rules all of the form,

$$\begin{aligned}&\text{NP} \rightarrow \text{NT}_1\ \text{NT}_2\ \text{NT}_3 \ldots \text{NT}_N\\ &\exists D_1 \quad : \quad D_1 \quad = \langle \text{NT}_1\text{AGREEMENT} \rangle \sqcup \langle \text{NT}_2\text{AGREEMENT} \rangle\\ &\exists D_2 \quad : \quad D_2 \quad = \langle \text{NT}_2\text{AGREEMENT} \rangle \sqcup \langle \text{NT}_3\text{AGREEMENT} \rangle\\ &\ldots \quad : \quad \ldots\\ &\exists D_{N-1} \quad : \quad D_{N-1} \quad = \langle \text{NT}_{N-1}\text{AGREEMENT} \rangle \sqcup \langle \text{NT}_N\text{AGREEMENT} \rangle\end{aligned}$$

and we define AGREEMENT to be the empty feature structure for the constituents or parts of speech, such as adverbs, that aren't declined to show intra-NP agreement, then this leads to a straightforward algorithm for testing the agreement of a phrase.

Assuming that we have first matched a rule of this form to our input, and that we have a lexicon from which we can build a list of lists of agreement feature structures (as in the example given earlier in figure 3.1), then figure 3.2 provides an algorithm for finding consistent interpretations. If the algorithm finds at least one consistent sequence then we assume that our noun phrase agrees. Otherwise we assume it does not.

In the description of the algorithm, we refer to our list of lists of agreement feature structures as a 'trellis' (though it will usually be a very sparse trellis). The algorithm uses a dynamic programming approach in which a second trellis, containing the same number of columns, is dynamically constructed. The columns of the second trellis are filled with pairs, each containing the unification of a consistent partial sequence and a list of corresponding feature structures defining the sequence up to that point. The idea is to remove the need to recompute unifications for common partial sequences, and to stop searching as soon as a 'dead end' is reached (if a unification fails then nothing is added to the second trellis's column).

As presented, the algorithm is more general and does more work than is strictly necessary for agreement checking. There are at least a couple of simple modifications that can be made if the performance should prove inadequate:

- Instead of storing partial sequences at each node of the unification trellis, the algorithm

```
Find-Consistent-Seqs(Trellis)
    Let N be the number of columns in Trellis
    Initialise U-Trellis to a list of N empty lists
    For each FS in Trellis[0]
        Let Seq be the sequence containing FS only
        Append (FS, Seq) to U-Trellis[0]
    End for
    For each Col in Trellis[i=1..N-1]
       For each FS in Col
            For each Pair in U-Trellis[i-1]
                If FS unifies with FS(Pair) Then
                    Let UFS be the unification of FS and FS(U-Entry)
                    Let Seq be the partial sequence Seq(U-Entry) + FS
                    Append (UFS, Seq) to U-Trellis[i]
                End if
            End for
        End for
    End for
    Initialise ConsistentSeqs to an empty list
    For each Pair in U-Trellis[N-1]
        Append Seq(Pair) to ConsistentSeqs
    End for
    Return ConsistentSeqs
```

Figure 3.2: Algorithm for finding consistent sequences within a ‘trellis’ of agreement feature structures.

could be modified to store back-pointers, from which full sequences could be reconstructed at the final step.

- Since we only need to determine whether or not a consistent sequence exists, we could use a variant that searches the trellis depth-first (if the first column is viewed as the 'top') and returns a result as soon as the first full consistent sequence is found. This would also remove the need to store any representation of the partial sequences.

3.4 Summary

We have proposed a simple unification-based method for testing agreement. The method is based on the assumption that we can define or learn an appropriate set of grammar rules and a lexicon for the task, problems that we have not yet discussed in detail.

Chapter 4

Developing the Agreement Checker

4.1 Overview

This chapter describes the implementation of the agreement checker proposed in the previous chapter and presents results for the News Commentary corpus. The initial implementation is straightforward, but yields a high failure rate on fluent text. Through an incremental process of analysis and lexicon refinement, the failure rate is reduced significantly and the checker is then applied to translation output. Based on the results the lexicon is processed further to improve failure detection.

4.2 Learning the Initial Grammar Rules

To inform the development of a set of grammar rules for the agreement checker, we first performed an analysis of the use of different noun phrase constructions in the News Commentary parallel corpus[1], which contains approximately 1.5 million words each of English, German, French, Spanish, and Czech.

We parsed the German portion of the corpus using BitPar[2], a freely-available probabilistic context-free grammar parser designed for efficiently parsing treebank grammars, together with the accompanying German-language package, which includes a grammar extracted from the Tiger treebank[3].

After parsing the corpus, we wrote a tool to process the BitPar output (which uses a Lisp-like parenthetical notation to represent parse trees). With heuristic 1 in mind, we used this tool to search for noun phrases containing exactly one common noun (a 'NN' in the STTS tagset used by the Tiger Corpus[4]) and extracted the sequence of part-of-speech tags at the bottom

[1] http://www.statmt.org/moses/?n=Moses.LinksToCorpora
[2] http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/BitPar.html
[3] http://www.ims.uni-stuttgart.de/projekte/TIGER/
[4] http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/stts.asc

Rank	POS Sequence	Freq.	Per.	Cum.	Example[a]
1	ART NN	40,550	33.44	33.44	*der Hund*
2	ART ADJA NN	18,388	15.16	48.60	*der große Hund*
3	ADJA NN	16,610	13.70	62.30	*große Hunde*
4	PPOSAT NN	6,355	5.24	67.54	*mein Hund*
5	PDAT NN	5,207	4.29	71.83	*dieser Hund*
6	PIAT NN	4,012	3.31	75.14	*alle Hunde*
7	NN	2,103	1.73	76.88	*Hunde*
8	PPOSAT ADJA NN	1,603	1.32	78.20	*mein großer Hund*
9	ART ADJA ADJA NN	1,371	1.13	79.33	*der große braune Hund*
10	ADJA ADJA NN	1,332	1.10	80.43	*große braune Hunde*
...	...	...	...	...	...
42	NN PROAV	134	0.11	90.03	-
...	...	...	...	...	...
221	NN CARD XY	13	0.01	95.00	-
...	...	...	...	...	...
3,992	$PAR $PAR ADJD VVPP $PAR VAFIN PDS ADV PPOSAT NN	1	0.00	100.00	-

[a]The examples translate as ‘the dog’, ‘the big dog’, ‘big dogs’, ‘my dog’, ‘this dog’, ‘all dogs’, ‘dogs’, ‘my big dog’, ‘the big brown dog’, and ‘big brown dogs’, respectively.

Table 4.1: Most Common single-NN noun-phrases in News Commentary corpus.

of the noun-phrase subtree. We counted the occurrences of each distinct sequence, of which table 4.1 shows the 10 most highly-ranked, together with their frequencies as absolute values, percentages, and cumulative percentages. The top 10 entries account for just over 80% of all single-NN noun phrases. The table also shows the 42nd entry, which marks 90% coverage, the 221st, which marks 95%, and the final entry, the 3,992nd.

In natural language processing, Zipf’s law is well-known for relating the frequency and rank of words and describing the ‘long-tail’ effect seen in language (a good discussion is provided in Manning and Schütze (1999, chapter 1)). As can be seen in the log-log plot given in figure 4.1, the noun phrase data doesn’t fit Zipf’s law exactly (since the law would predict the data should follow the line with gradient -1) but it does appear to follow a similar power distribution.

For the purposes of this dissertation, we are only interested in obtaining an approximate picture of intra-noun phrase agreement failure rates and so we chose to base our grammar

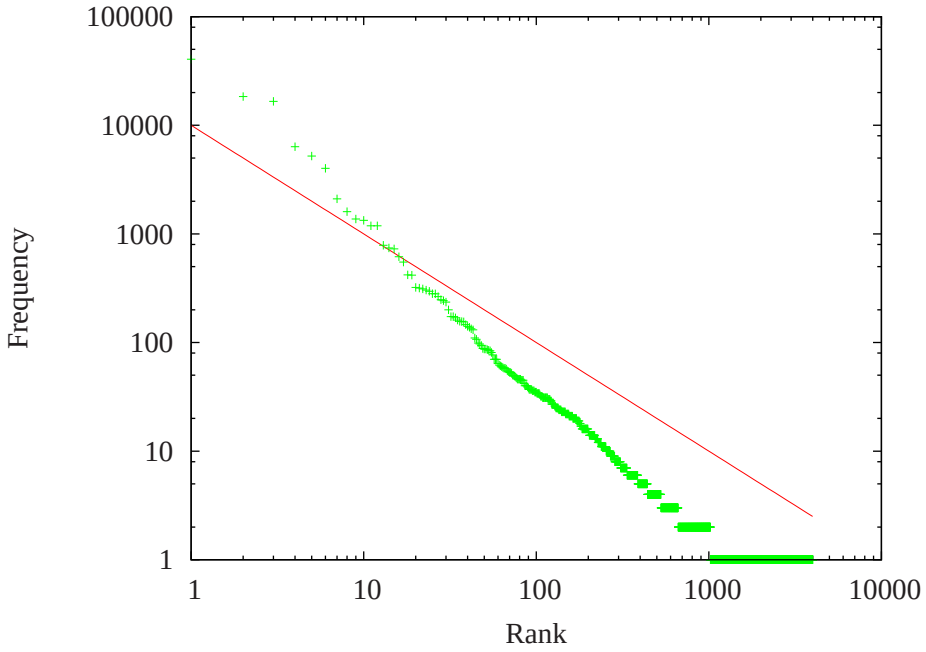


Figure 4.1: Log-log plot of rank vs frequency for single-NN noun phrases in the News Commentary corpus. The line has gradient -1.

rules on the top 42 entries, or 90% of occurences. Though note that this is likely to give an underestimate since longer phrases tend to be rarer and can be expected to contain more agreement errors since they are less likely to be translated as complete phrases.

Since intra-NP agreement is marked only by determiners and adjectives, we pruned this list to those that contain at least one of either, which leaves 32 rules. This gives us the initial set of grammar rules for the agreement checker, shown in figure 4.2. All rules produce a noun phrase so the left-hand side is omitted, as are the unification identities between the agreement feature structures of the non-terminals. Since the rule set is so small, and to simplify the implementation, we didn't decompose the rules into constituents (for example, to generalise the category of a determiner) but left the rules as 'flat' part-of-speech sequences.

4.3 Learning the Initial Lexicon

In addition to producing parse trees with syntactic categories and part-of-speech tags, BitPar labels nodes with morphological information. The following subtree shows a noun phrase from the parsed News Commentary corpus:

```
(NP-CJ
  (PPOSAT-NK-Dat.Sg.Fem seiner)
  (NN-NK-Dat.Sg.Fem Wiederwahl))
```

ART NN	ART PIAT NN	ART $Par NN $Par
ART ADJA NN	PDAT ADJA NN	ADJA KON ADJA NN
ADJA NN	ADV ADJA NN	PRELAT NN
PPOSAT NN	ADV ART ADJA NN	PWAT NN
PDAT NN	ART ADJD ADJA NN	ART $Par ADJA NN $Par
PIAT NN	ART NN PROAV	ART ADJA KON ADJA NN
PPOSAT ADJA NN	ART ADV ADJA NN	CARD ADJA NN
ART ADJA ADJA NN	ADJD ADJA NN	KOKOM ART ADJA NN
ADJA ADJA NN	KOKOM ART NN	ART NN CARD
PIAT ADJA NN	ART NN ADV	ART TRUNC KON NN
ADV ART NN	ADV PIAT NN	

Figure 4.2: The 32 POS sequences selected for the checker's grammar.

It contains a possessive adjective, *seiner*, and a common noun, *Wiederwahl*, which are both labelled as dative, singular, and feminine. Being probabilistic in nature, the results are occasionally inconsistent:

```
(NP-OA
  (PPOSAT-NK-Acc.Sg.Fem seine)
  (NN-NK-Acc.Sg.Masc Wiederwahl))
```

(Here the gender of *Wiederwahl* is incorrectly labelled *Masc*.)

To learn the initial lexicon we will simply extract the words and their feature information from a BitPar parse of the training data, but will bear in mind that if we extract the feature information as-is then our lexicon will contain spurious entries introduced at some unknown error rate. The presence of spurious feature structures is likely to cause our recogniser to miss genuine agreement errors by finding a consistent interpretation when none should exist. However low the BitPar error rate may be, we should note that the larger the corpus from which we extract our lexicon, the higher we can expect the probability to be of encountering a spurious feature structure for a given word.

4.4 Implementation

Based on the method proposed in chapter 3, and with the initial set of grammar rules and lexicon as just described, we implemented an agreement checker using the Python programming language.

Both Shieber (1986) and Jurafsky and Martin (2008) describe the implentation of feature structures as directed acyclic graphs, but since we don't make use of reentrancy and the infor-

mation we wish to represent is so minimal, we simply used a `dict`, Python's built-in mapping type, to store a flattened representation of our features.

Our unification operator then simply has to check that if a key is present in two feature structures then it has the same value in both. The result, if unification is successful, is a third `dict` containing the union of the two operand's feature-values pairs. If unification fails, the failure is communicated to the caller.

The `Find-Consistent-Seqs` algorithm was implemented much as in the pseudocode given earlier in figure 3.2.

The checker is presented with the full translation output and must isolate noun phrases in order to construct the corresponding agreement trellises and apply the `Find-Consistent-Seqs` algorithm. It scans the input from left to right, at each step searching the rule set for matches based on the part-of-speech features. The rules are tested in order of length, longest first. For example, if the input contains a sequence of words consisting of an article, an adjective, and a noun, then it will apply the longer ART ADJA NN rule in preference to the ADJA NN rule, the idea being that the more words are included, the better the chance of detecting an agreement failure.

4.4.1 Extracting the Lexicon

Since we will later train a machine translation system using this corpus, and since when preparing the training data we will drop sentences longer than a given length (40 tokens in this case), we also rejected those sentences from the lexicon extraction process.

From the parsed corpus we extracted the lexicon using a simple Python program based on the BitPar output-parsing tool that we developed to perform the noun-phrase analysis described in section 4.2. The program extracted words together with their part-of-speech, case, gender, and number features, as given by BitPar (omitting indeterminate features for which BitPar uses the wildcard '*'). The program also adds a count for each entry, which whilst not used by the recogniser is useful for later analysis.

The process is straightforward and the only point worth noting is that of file encoding: BitPar expects input to use the ISO-8859-1 character set, a widely-used encoding for European text. Since the News Commentary corpus is encoded as UTF-8 we converted it using the standard POSIX program `iconv`, rejecting any sentence that contained one or more unconvertible characters.

BitPar was able to parse about 97.4% of the converted sentences. We rejected the remainder, leaving a corpus of 69,585 parsed sentences.

4.5 Results for the Training Data

To test the agreement checker, we first ran it on the News Commentary training data. Since the text was written by fluent German-speakers we can expect the true error rate to be very close to zero, and so this test will give an indication of our checker's false positive rate.

Since we already had part-of-speech tags, we tagged the input to aid the recognition of noun phrases.

It's worth commenting here on the fact that we are testing our checker on the same data from which its grammar rules and lexicon were learned, an approach that might seem dubious (and that is rightly avoided in statistical parsing, for example). However, note that the translation output we ultimately intend to check will use (a subset of) the same words[5] with the only differences from this test being that the translation system will have arranged those words in different orders and potentially with a different distribution. The former difference is exactly the one we're interested in; the latter we assume will not be significant on average. Note also that we assume the choice of rule set is neutral with respect to the phrase ordering a decoder produces.

If, as later, we wish to check the output of a translation system trained on different data then we will have to extract a new lexicon from that system's training data (and in that case we can't directly compare the results obtained here).

4.5.1 Initial Results, or Revision 0

On the first run, the checker recognised 224,317 noun phrases (an average of 3.2 per sentence) and calculated an overall agreement failure rate of 2.72%. Clearly this is far higher than we would expect in fluent text and an examination of the failed phrases immediately revealed two potential sources of spurious failures, both in the lexicon:

1. Plural forms have gender values. These are produced by BitPar, presumably to provide the gender of the corresponding singular forms, but they add a grammatical feature that is not involved in the declension of the surface forms.

2. Nouns have case values. Whilst in German nouns do mark case in a few limited contexts, in most they do not.

In principle, neither of these necessarily poses a problem for this test: provided BitPar assigns consistent genders to the majority of agreeing plural determiners, adjectives, and nouns, our lexicon should usually contain the entries it needs to find consistent agreement sequences. Similarly for case. However, the inclusion of these features will be problematic when we try to

[5] There are some subtleties to this, as we'll discover.

check translation output since many of the noun phrases we encounter will have been stitched together from phrases originating in multiple larger phrases.

4.5.2 Revision 1

We chose therefore to mask out the GENDER feature wherever the number is singular and mask out the CASE feature for nouns. To accomplish this we wrote a separate program to process the lexicon and generate an edited version. After re-running the checker with the new lexicon, we received a reduced failure rate of 1.51%.

Examining the remaining failed phrases revealed a high proportion of failures where a valid interpretation for an adjective was absent from the lexicon. For example, the training data contains this phrase

der|*ART schlüssige*|*ADJA und*|*KON direkte*|*ADJA beweis*|*NN*[6]

The noun *Beweis* is singular and masculine, and correctly appears as such in the lexicon. Based on this and on the declension of the article *der* and the two adjectives, the only possible case for this phrase is nominative. However our lexicon does not contain a nominative, masculine entry for *schlüssige*. Examining the parsed training data reveals that the case of that adjective has been incorrectly labelled in this particular occurrence,

```
(NP-SB
  (ART-NK-Nom.Sg.Masc der)
  (CAP-NK
    (ADJA-CJ-Pos.Acc.Sg.Fem schlüssige)
    (KON-CD und)
    (ADJA-CJ-Pos.Acc.Sg.Fem direkte))
  (NN-NK-Nom.Sg.Masc Beweis))
```

and that *schlüssige* appears only six other times in the corpus, never in the exact context we require: nominative, masculine, and singular.

4.5.3 Revision 2

Fortunately, German adjective declension is highly regular, as described earlier, and with few exceptions it is possible to infer an adjective's missing feature structures based on its suffix. If we see an adjective ending *-e*, like *schlüssige*, then we can infer the existence of seven agreement feature structures including the one required here:

$$\textit{schlüssige} \longmapsto \begin{bmatrix} \text{POS} & \textit{adja} \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & \textit{nom} \\ \text{GENDER} & \textit{masc} \\ \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix}$$

[6] In the corresponding English sentence this phrase is, 'the [most] convincing and direct proof.'

And similarly for adjectives ending *-em*, *-en*, *-er*, and *-es*.

Adding this ability to our lexicon processor and re-running, the checker gives a new failure rate of 0.17%, or 382 out of 224,317 recognised noun phrases.

4.5.4 Revision 3

In analysing the agreement failures we noticed a further problem with our lexicon; in fact, a general problem with the approach that we'd not fully appreciated earlier: Among the thousands of occurrences of a common word, BitPar — as will be inevitable for any probabilistic language processing tool — will assign a handful of incorrect features, which are extracted and included in the lexicon. And it will occasionally omit features resulting in incomplete lexicon entries. In the case of determiners, a small closed set of frequently-occurring function words, the additional presence of incorrect or incomplete entries is likely to cause our checker to miss significant numbers of genuine agreement failures.

The effect is most striking for those words at the peak of the Zipf-ian distribution, which, of the words we're interested in — determiners, adjectives, and nouns — will certainly include the articles. Inspecting the entries for the articles revealed that only two of the six definite articles had perfect entries, whilst the other four also had incomplete or erroneous entries. Similarly, for the indefinite articles, only two out of six had perfect entries.

These entries result from comparatively small numbers of morphological labelling errors. For example, the parsed News Commentary corpus contains 5,554 occurrences of the definite article, *dem*, of which 3,622 are assigned the following features,

$$dem \longmapsto \begin{bmatrix} \text{POS} & art \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & dat \\ \text{GENDER} & masc \\ \text{NUMBER} & sg \end{bmatrix} \end{bmatrix}$$

and 1,841 are assigned the features

$$dem \longmapsto \begin{bmatrix} \text{POS} & art \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & dat \\ \text{GENDER} & neut \\ \text{NUMBER} & sg \end{bmatrix} \end{bmatrix}$$

Both of these are correct and are the only two valid feature structures for this article. However, we also encounter this incomplete entry 22 times,

$$dem \longmapsto \begin{bmatrix} \text{POS} & art \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & dat \\ \text{NUMBER} & sg \end{bmatrix} \end{bmatrix}$$

Here, the absence of a gender feature will cause our checker to miss agreement errors where this definite article is used with a female, singular noun in a dative context. Additionally, we encounter this incorrect entry 66 times,

$$dem \longmapsto \begin{bmatrix} \text{POS} & art \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & acc \\ \text{GENDER} & masc \\ \text{NUMBER} & sg \end{bmatrix} \end{bmatrix}$$

which would cause the checker to miss the failure in, for example, *dem braune Hund*. Worst of all, we encounter the following degenerate entry three times:

$$dem \longmapsto \begin{bmatrix} \text{POS} & art \\ \text{AGREEMENT} & [] \end{bmatrix}$$

There are some obvious statistical approaches to deciding which feature structures are likely to be correct and which are not. However, given the small number of article and possessive adjective forms (for the articles: 12 distinct surface form words requiring a total of 28 feature structure entries; for the possessive adjectives: 33 distinct surface form words and 85 feature structures) and given their high occurrence among the determiners (as seen earlier in table 4.1 and figure 4.2), we decided to hand-write the lexicon's article and possessive adjective entries and discard those we had extracted.

It's worth nothing though that the benefit of a statistical approach will increase with the size of the corpus, as the numbers of content word occurrences grows (and thus so does the number of words with spurious entries).

We added the facility to our lexicon processor to read entries from a separate file and override those in the lexicon. After hand-writing a complete set of *art* and *pposat* entries and overriding the lexicon entries, we re-ran the checker, this time receiving an agreement failure rate of 0.39%, an increase as would be expected.

4.5.5 Revision 4

Satisfied that the most glaring lexicon problems had been removed, we performed a per-rule analysis of agreement failure rates. Our original rule selection was made purely on the basis of an empirical analysis of the corpus and we did not consider the specific rules except to remove those not containing either a determiner or adjective.

Table 4.2 shows the rules in order of application frequency and shows the individual failure rates. The high failure rates of a few rules particularly stand out and clearly need examining:

ART PIAT NN

This rule has an agreement failure rate of 10.84%. Examples from the data are

Rule	Uses	Failures	Rule	Uses	Failures
ART NN	88,125	0.24%	ART PIAT NN	1,255	10.84%
ART ADJA NN	39,215	0.59%	ART ADJD ADJA NN	1,071	1.49%
ADJA NN	32,622	0.09%	PDAT ADJA NN	1,054	0.09%
PPOSAT NN	10,706	0.15%	ADV PIAT NN	1,014	0.20%
PDAT NN	8,284	0.04%	ART ADV ADJA NN	754	1.72%
PIAT NN	7,099	0.13%	KOKOM ART NN	732	0.41%
ADV ART NN	4,452	0.20%	ART ADJA KON ADJA NN	719	1.67%
ART NN ADV	3,804	0.34%	ART NN PROAV	686	0.29%
ADJA ADJA NN	3,690	0.89%	CARD ADJA NN	558	0.18%
ART ADJA ADJA NN	3,056	0.75%	ART NN CARD	538	0.37%
PPOSAT ADJA NN	3,019	0.53%	ART $PAR NN $PAR	514	2.14%
PIAT ADJA NN	2,427	0.16%	PRELAT NN	316	8.23%
ADV ADJA NN	2,291	0.26%	ART $PAR ADJA NN $PAR	282	1.42%
ADV ART ADJA NN	2,169	0.97%	ART TRUNC KON NN	278	3.24%
ADJD ADJA NN	1,783	0.28%	PWAT NN	266	0.00%
ADJA KON ADJA NN	1,290	0.23%	KOKOM ART ADJA NN	248	0.81%

Table 4.2: Agreement failure rates per grammar rule on the training data.

ein|ART *paar*|PIAT *Wochen*|NN
ein|ART *wenig*|PIAT *Hintergrundinformation*|NN[7]

In phrases of this form, the article agrees with the quantifier, not the noun, so this rule should be removed.

PRELAT NN

This rule has an agreement failure rate of 8.23%. Examples from the data are

dessen|PRELAT *Stabilität*|NN
dessen|PRELAT *Krönung*|NN[8]

In phrases of this form, the relative pronoun refers to and agrees with an earlier subject, so this rule should also be removed.

Rather than make grammatical judgements on all of the remaining 30 rules, we decided on a blanket removal of all rules with an agreement failure rate above 1% on the training data. These eight rules account for 2.3% of the noun phrases checked, yet account for 25.9% of the agreement failures. In some instances, the removal of a longer rule, such as ART $PAR ADJA NN $PAR, will allow a shorter rule to be applied (ADJA NN, in this case).

[7] 'A few weeks' and 'a little background information'
[8] 'whose stability' and 'whose coronation'

Re-running the checker with the remaining 24 rules gives a failure rate of 0.29%.

4.5.6 Interim Summary

The following table shows a summary of the lexicon and rule set processing performed so far, and shows the corresponding agreement failure rates on the News Commentary training data.

Revision	Grammar Processing	Failure Rate
0	None	2.72%
1	Remove CASE if *nn*, GENDER if *sg*	1.51%
2	+ Infer missing *adja* entries	0.17%
3	+ Override *art*, *pposat* entries	0.39%
4	+ Remove suspect rules	0.29%

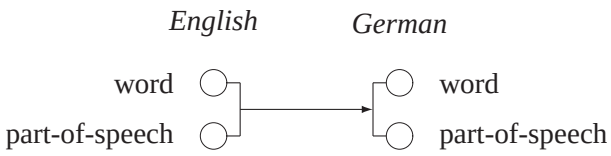
For now, the false positive rate seems low enough for us to begin applying the checker to translation output.

4.6 Results for Translation Output

In this section we apply our agreement checker to the output of a statistical machine translation system trained on the data from which we extracted our lexicon and derived our rules, the News Commentary corpus.

4.6.1 Methodology

Our agreement checker was tested using text tagged with part-of-speech information and by using Moses and a factored translation model our SMT system can also produce tagged output. Following the diagram style of Koehn and Hoang (2007), our factored model is as follows:



That is to say, there are two factors, the surface form words and the part-of-speech tags, on both the source and target sides. The word alignment and phrase extraction processes operate on part-of-speech-tagged words as if they were single entities, resulting in a single translation table containing entries such as the following,

```
[X] [X] ||| a|dt watchdog|nn ||| einem|art wachhund|nn ||| ...
```

We use Moses's chart-parsing decoder with a hierarchical phrase-based rule set.[9]

Training more or less follows the procedure described for the baseline system of the translation task from the EACL 2009 Fourth Workshop on Statistical Machine Translation[10] (WMT 09). The Moses manual describes the typical training process in detail.[11]

The main difference with our factored model is that we must tag the words. For the English text we used the tagger from the MontyLingua NLP Toolkit,[12] which is based on the Brill tagger. For the German text, we extracted part-of-speech tags from the BitPar output. Where either conversion to UTF-8 or parsing failed, we recorded the line numbers and removed the corresponding sentences from the English text. Since MontyLingua successfully tagged all sentences, the equivalent filtering was not necessary. Our adapted training procedure is shown in figure 4.3.

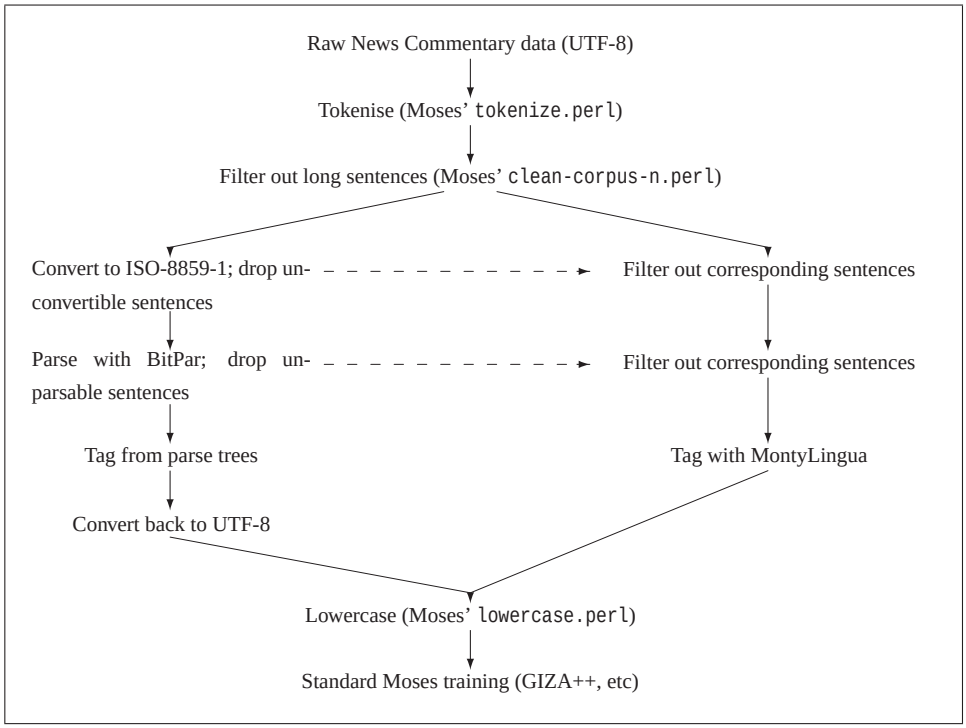


Figure 4.3: The training process for our factored model. The left and right branches show German- and English-specific steps, respectively.

We built three 5-gram language models using the SRILM toolkit[13] and trained on the following data:

[9] This choice was made early on when we anticipated extending the grammar. We could alternatively have used a standard phrase-based model.

[10] `http://www.statmt.org/wmt09/translation-task.html`

[11] `http://www.statmt.org/moses`

[12] `http://web.media.mit.edu/~hugo/montylingua`

[13] `http://www.speech.sri.com/projects/srilm/`

Source	Sentences
News Commentary	82,740
Europarl	1,418,115
News Train 08	10,193,376

We then interpolated them to produce a single mixed-source language model.

Whilst training the translation model on the News Commentary corpus will produce a tiny SMT system by current research standards — Europarl, at 1.4 million sentence-pairs, is approximately 17 times larger, whilst the GigaWord French-English corpus, for example, is over 270 times larger, running to 22 million sentence-pairs — we wanted to use a realistically-sized language model to reduce the chance of a misleadingly high agreement failure rate due to using a weak SMT system. The assumption here is that the generally broader coverage of the language model will be a bigger factor in producing intra-noun phrase agreement than will the translation model. Whether or not this assumption is sound, we will be training a larger system in the next chapter.

Tuning followed Moses' standard Minimum Error Rate Training (MERT)-based procedure (Och, 2003), finding weights for the language model, translation table scores, and word penalty. We used the 1025-sentence 'news-dev2009a' tuning data from the WMT 09 translation task. The English input text was tagged, as during training. We configured Moses to output the surface-form factor only, so the German reference text did not require part-of-speech tags. For this purposes of this experiment, we didn't explore the effect of tuning against tagged reference data. Nor did we develop a part-of-speech n-gram model, an addition that has been shown to improve translation quality (Koehn and Hoang, 2007).

After tuning, Moses was run on the 1026-sentence 'news-dev2009b' development test set. By re-configuring to include the part-of-speech output factor we obtained the first set of translated test sentences for our agreement checker.

We then repeated the above process using a second set of tuning and test data: the 2000-sentence 'dev2006' and 'devtest2006' sets, taken from a previous year's WMT translation task.

4.6.2 Initial Results

We ran the checker over both sets of tagged translation output using the lexicons and rules sets from all testing revisions that were described in section 4.5. The results are shown in table 4.3.

The results for past revisions were produced mainly to satisfy ourselves that our grammar processing steps don't have a disproportionate effect on results for translation output.

The results we are most interested in are those for our latest lexicon and rule set, for which we received an error rate of 0.29% on the training data, and from which we believe we should see the fewest false negatives and false positives.

For the 'news-dev2009b' output we received an agreement failure rate of 5.28%, or 151 failures out of 2,858 recognised noun phrases. The first ten failures are shown here, in boldface,

Lexicon Processing	**Training**	**Test 1**	**Test 2**
None	2.72%	6.90%	4.21%
Remove CASE for *nn*, GENDER for *sg*	1.51%	3.76%	2.64%
+ Infer missing *adja* entries	0.17%	2.54%	1.53%
+ Override *art*, *pposat* entries	0.39%	5.57%	3.94%
+ Prune rule set	0.29%	5.28%	3.46%

Table 4.3: Failure rates determined by the agreement checker for the training data and for the 'news-dev2009' (Test 1) and 'devtest2006' (Test 2) translation output.

together with line numbers and a few words of surrounding context. We omit the part of speech tags in favour of showing more context:

4 *nach* ***einer analyst*** *, der ezb ist in* ***einer catch 22*** *: es muss . . .*
10 *. . . 2 prozent unterstützung jedes innerhalb* ***der gesamten beispiel*** *.*
13 *13 prozent* ***der beispiel*** *sagte sie vertrautenf oder sehr vertrauten . . .*
15 *. . . einem referendumn , 60 prozent* ***der beispiel*** *sagte es jedenfalls würde . . .*
23 *. . . einer flasche mit* ***einer vorkommen*** *auf sie , so der . . .*
37 *. . . wurden nach* ***einem schweren verkehrsunfällen*** *zufall . . . .*
44 *. . . , dass* ***die gegenwärtigen krankenhaus*** *bedingungen sind nicht förderlich . . .*
55 *. . . auf friday , receipt* ***der professionelle meinung*** *. die meinung . . .*
56 *. . . war für* ***seine eigenen auto*** *in seinem bezirk . . . .*

Note that the nouns in the first two phrases are not untranslated words. The first, *analyst*, happens to be the same word as in English. The second, *catch*, is not a German word, but does appear as part of the borrowed phrase *"Catch 22"* in the News Commentary text.

Nonetheless, untranslated words do pose a problem. Our checker implicitly handles them by ignoring words that do not appear in the lexicon when constructing the trellis.

For the 'devtest-2006' output we received an agreement failure rate of 3.46%, or 268 failures out of 7,744 recognised noun phrases.

There is a considerable difference between the failure rates for the two sets of translation output. However, both test sets exhibit significantly higher agreement failure rates than the fluent test data and have been demonstrated to detect genuine agreement failures. There are still deficiencies in the lexicon that we know cause the checker to miss failures and so we attempt to reduce them in two further revisions.

4.6.3 Revision 5

The declension of an attributive adjective depends upon case, gender, and number, but as was briefly mentioned in section 2.2.3, also depends upon the presence and type of determiner used in the noun phrase. When a determiner with an expressive ending, such as the definite article, is used, an adjective will take one of only two suffixes, *-e* and *-en*, and this is called weak declension. When no article is present, the adjective takes one of six more-expressive suffixes, and this is called strong declension. For other determiners, the so-called 'mixed' declension is used.

The following table show the suffixes used in the weak, mixed, and strong declensions for the nominative case:

Gender/Number	**Weak**	**Mixed**	**Strong**
Masculine	*-e*	*-er*	*-er*
Feminine	*-e*	*-e*	*-e*
Neuter	*-e*	*-es*	*-es*
Plural	*-en*	*-en*	*-e*

For example, when the definite article is used with a singular neuter noun, an adjective will carry the suffix *-e*:

das grüne Krokodil

whereas when the indefinite article is used, an adjective will carry the suffix *-es*:

ein grünes Krokodil

Our current grammar does not account for this variation and so our checker cannot detect the agreement failure in,

* *ein grüne Krokodil*

As already mentioned, adjective declension is highly regular in German. The few special cases include some idioms originating in older German and adjectives ending in *-er* that derive from town names and numerals (Durrell, 2002, sec. 6.2.7).

We can exploit this regularity and encode a rule for declension in our lexicon by augmenting our agreement feature structures with an additional feature, DECLENSION. For the determiners it will variously take the values *weak* and *mixed*. For adjectives the value will indicate a possible declension type according to the suffix in combination with the case, gender, and number. For example, the nominative singular neuter entry for *grüne* will become

$$\textit{grüne} \longmapsto \begin{bmatrix} \text{POS} & \textit{adja} \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & \textit{nom} \\ \text{DECLENSION} & \textit{weak} \\ \text{GENDER} & \textit{neut} \\ \text{NUMBER} & \textit{sg} \end{bmatrix} \end{bmatrix}$$

The agreement test for *ein grüne Krokodil* will now fail due to the combined presence of the *mixed* value for the indefinite article, *ein*, such as in,

$$ein \longmapsto \begin{bmatrix} \text{POS} & art \\ \text{AGREEMENT} & \begin{bmatrix} \text{CASE} & nom \\ \text{DECLENSION} & mixed \\ \text{GENDER} & neut \\ \text{NUMBER} & sg \end{bmatrix} \end{bmatrix}$$

and the absence of any compatible lexicon entry for *grüne* with which to unify.

Our lexicon processor already overrides the articles and possessive adjectives, so we modified our hand-written entries to include the DECLENSION feature with the value *weak* for the definite articles and *mixed* for the indefinite articles and possessive adjectives (we omitted the other determiners, which are both less-commonly occurring and have more numerous surface-forms).

For adjectives, we extended the lexicon processor to add a DECLENSION feature with the appropriate value based on the suffix of the word's surface form. Where necessary, the processor generates additional entries (for example, *grünes* is the singular neuter nominative form for both the mixed and strong declensions, so requires distinct feature structure for each).

After implementing this and processing the last revision of our lexicon, we re-ran the checker on our training data and the two sets of translation output. On the training data, the failure rate increased from 0.29% to 0.38%. On the 'news-dev2009b' translation output, it increased from 5.28% to 5.81%. On the 'devtest2006' translation output, it increased from 3.46% to 3.96%.

4.6.4 Revision 6

As we have already seen, our lexicon extraction method can introduce both incomplete and erroneous feature structures. With the inference of missing feature structures for the attributive adjectives (revision 2), we ensure our lexicon will contain a complete set of feature structures for every regular adjective. We therefore tried removing any incomplete, adjective feature structure (and left the complete-but-irregular entries in the hope of catching most of the special cases), the idea being that this might result in a lower false positive rate.

As usual, we re-ran the checker on our training data and the two sets of translation output. On the training data, the failure rate increased from 0.38% to 0.63%. On the 'news-dev2009b' translation output, it increased from 5.81% to 6.16%. On the 'devtest2006' translation output, it increased from 3.96% to 4.16%.

The closeness of failure rate increases between the training and test sets suggests that this change is probably introducing more false negatives than genuine agreement failures.

4.7 Summary

We implemented a simple intra-noun phrase agreement checker based on the approach proposed in chapter 3. We used a small empirically-derived rule set and a lexicon extracted automatically from parsed training data. Using the raw agreement feature values we saw an agreement failure rate of 2.72% on our training data, which gives an estimate of the false positive rate for translations derived from the same data, and saw agreement failure rates of 6.90% and 4.21% on the output of a hierarchical phrase-based machine translation system trained on the same data.

By removing grammar rules that produced high failure rates on the training data, and by applying a small number of broad transformations to the lexicon aimed at reducing both the false positive and the false negative rates, we achieved (at revision 5) an agreement failure rate of 0.38% on the training data and 5.81% and 3.96% on our two sets of translation output. Whilst the results obtained so far look promising, we note that the News Commentary training corpus is considerably smaller than would be used in a state-of-the-art-system, and so these failure rates may be misleadingly high.

Table 4.4 shows the full set of News Commentary results.

Revision	**Description**	**Training**	**Test 1**	**Test 2**
0	None	2.72%	6.90%	4.21%
1	Remove CASE for *nn*, GENDER for *sg*	1.51%	3.76%	2.64%
2	+ Infer missing *adja* entries	0.17%	2.54%	1.53%
3	+ Override *art*, *pposat* entries	0.39%	5.57%	3.94%
4	+ Prune rule set	0.29%	5.28%	3.46%
5	+ Add DECLENSION feature	0.38%	5.81%	3.96%
6	+ Remove incomplete regular *adja* entries	0.63%	6.16%	4.16%

Table 4.4: Failure rates determined by the agreement checker for the training data and for the ‘news-dev2009’ (Test 1) and ‘devtest2006’ (Test 2) translation output.

Chapter 5

A Feature Function for Agreement

5.1 Overview

This chapter describes the integration of the agreement checker into a machine translation system. The checker is implemented as a feature function in a log-linear hierarchical phrase-based model. The translation system is trained on the Europarl corpus, which at 1.4 million sentence-pairs is substantially larger than the News Commentary training corpus used in section 4.6.

We first train a baseline system — an otherwise-identical system without the new agreement feature function — and use our checker to obtain agreement failure rates on the output. After implementing the feature function we first use it as a 'hard' constraint: that is, we set a sufficiently high penalty for agreement failure that any failing hypothesis will be rejected. We then assign the feature function a tunable weight and re-tune the system using the standard MERT process.

5.2 The Baseline Translation Model

The baseline translation system is identical to that used in section 4.6 to develop the agreement checker, except that here we use the much larger Europarl corpus as training data. The language model, which we earlier speculated would be the more significant factor of the two in producing translation agreement, is unchanged.

As before, the training procedure uses BitPar and MontyLingua to tag the German and English data respectively, and follows the same steps shown in figure 4.3.

For tuning and testing, we re-used the 'news-dev2009a' and 'news-dev2009b' data sets. Even after filtering for this input, our hierarchical phrase-based grammar still contains tens of millions of rules and proves impractical to decode with, so we prune it to the most probable 100 translation options per phrase according to $p(t|s)$.

Lexicon Processing	Training	Translation
None	0.58%	3.45%
Remove CASE for *nn*, GENDER for *sg*	0.37%	2.20%
+ Infer missing *adja* entries	0.05%	1.35%
+ Override *art*, *pposat* entries	0.12%	3.88%
+ Add DECLENSION feature	0.16%	4.41%
+ Remove incomplete regular *adja* entries	0.31%	4.90%

Table 5.1: Failure rates determined by the agreement checker for the Europarl training data and for the 'news-dev2009' translation output.

5.3 Agreement Results

We first use our agreement checker to measure failure rates for the tagged Europarl training data and for translations produced by the baseline system. If our checker can find little difference between the failure rate on the fluent training data and on the translation output then there is probably little value in implementing a feature function to enforce agreement as determined by the same method.

We extracted a lexicon from the parsed training data using the program developed in chapter 4. To the raw lexicon, we applied the same process of transformations as for the News Commentary lexicon (revisions 1–4 and revision 6). We did not repeat the process of rule derivation, and re-used the reduced set of 24 rules from revision 5, assuming that the distribution of single-NN noun phrase will be similar within the two corpora.

For the training data, our checker identifies and tests 3,213,099 noun phrases. For the translation data, it identifies and tests 3,040. Table 5.1 shows the results. They cannot be directly compared to the News Commentary results of table 4.4, since we now use the reduced rule set throughout. However a few differences are worth commenting upon:

- Agreement failure rates are considerably lower on both training and translation data. Whilst the reduction in translation agreement failures could be explained as resulting from a stronger translation model, the corresponding reduction in failure rates on fluent data make this explanation much less convincing.
- The final lexicon processing step — the removal of incomplete regular *adja* — now appears to have a greater effect on translation than training results. However, this step is still questionable for the reasons given earlier.

Based on these results, the translation failure rate appears sufficiently high that an approach to reducing it is worth pursuing. It's likely also that further lexicon processing would increase

failure detection further. For example, we have not investigated the effect of spurious noun entries with incorrect or missing gender values.

5.4 Developing and Integrating the Feature Function

Developing the agreement feature function was straightforward since the general procedure for feature function development and integration is described in the Moses manual[1]. There were some minor differences as the code of the chart-parsing branch has diverged, but nothing worth describing here.

The feature function is implemented as a subclass of Moses' `ScoreProducer` class, which underlies all feature functions, including the language and translation models. Our subclass, `AgreementScoreProducer`, defines a `CalcScore` member function, which is called to score every hypothesis produced during decoding. The function is essentially a rewrite in C++ of our Python agreement checker. It scans a hypothesis's part-of-speech factors from left to right whilst searching its ruleset, from largest to smallest, looking for applicable rules.

On finding a match, `CalcScore` looks up the corresponding surface-form words in the lexicon and builds a trellis of their agreement feature structures. The trellis is searched using an implementation of the `Find-Consistent-Seqs` algorithm and if no consistent sequence is found then the phrase is deemed not to agree.

For every agreement failure, a fixed penalty of 1 is subtracted from the hypothesis' log probability score, which initially is 0. The feature function's weight is managed independently by Moses.

Externally, this weight is configured through a new parameter, `weight-a`. This is set by the MERT tuning script and at the end of decoding the agreement scores are communicated back to the script in the n-best list.

5.5 Agreement as a Hard Constraint

We first tested the feature function as a hard constraint. That is, one that is sufficiently heavily-weighted that it is impossible, or at least highly-improbable, that a non-agreeing hypothesis will score higher than one that agrees. We arranged this by setting `weight-a` to 1 and leaving all other weights unchanged from those of the baseline. The feature function's penalty was left at -1. Since the other weights are normalised to sum to 1, the agreement feature function should almost always outweigh them.

Whilst hoping to fix agreement problems in the system's translation output, we also hope — somewhat speculatively — that the early removal of non-agreeing hypotheses will free up

[1] `http://www.statmt.org/moses`

some of the decoder's hypothesis stack space to allow a wider range of translation alternatives, rather than many morphological variations of the same few. And if applied during tuning, may increase diversity in the n-best lists.

With this configuration, we re-ran Moses twice, once configured to produce part-of-speech factors and once without.

Running the standalone agreement checker on the tagged output confirmed an agreement failure rate of exactly 0%. We then evaluated the surface-form output as per the baseline. The BLEU score was 12.27, a small increase on the baseline of 12.19. The NIST score was 4.7859, a small increase on the baseline of 4.7824.

5.5.1 Analysis

A benefit of using a narrowly-focussed feature function is that by re-using the baseline weights, it's possible to perform a meaningful word-by-word comparison of the before and after translations. The only differences should be those introduced by the agreement feature function. In some cases these will be difficult to interpret since the rejection of a non-agreeing hypothesis may have effects reaching beyond the noun phrase, but in many cases, as is our intention, the changes are localised to the non-agreeing words.

An inspection of these differences reveals a few common types of change:

Declensional Changes

These are the changes we are hoping to see: the root morphemes are unchanged, but determiner or adjective declension is altered so that the phrase agrees. This is the single most common type of change introduced. Examples from our evaluation set include,

BEFORE	…	*nicht*	*zu*	***einem***	***sensiblen***	***seite***	*:*	*sie*	*ist*	…
		PTKNEG	PTKZU	ART	ADJA	NN	$.	PPER	VAFIN	
AFTER	…	*nicht*	*zu*	***einer***	***sensiblen***	***seite***	*:*	*sie*	*ist*	…
		PTKNEG	PTKZU	ART	ADJA	NN	$.	PPER	VAFIN	

and

BEFORE	***die***	***langen***	***wochenende***	*mit*	*einem*	*preis*	…
	ART	ADJA	NN	APPR	ART	NN	
AFTER	***das***	***lange***	***wochenende***	*mit*	*einem*	*preis*	…
	ART	ADJA	NN	APPR	ART	NN	

both of which are reasonable. Unsurprisingly, the changes do not always produce declension appropriate to the context, as in the following example:

BEFORE	*(*	*in*	*2003*	*,*	***die***	***gleichen***	***korb***	*kam*	*,*	…
	NN	APPR	CARD	$,	ART	ADJA	NN	VVFIN	$,	
AFTER	*(*	*in*	*2003*	*,*	***den***	***gleichen***	***korb***	*kam*	*,*	…
	NN	APPR	CARD	$,	ART	ADJA	NN	VVFIN	$,	

Here the internal agreement is fine, but the noun phrase should use the nominative case since it is the subject[2]. Accounting for case is of course a separate (and more difficult) problem.

Alternative Word Choice

Less often (we quantify these changes later), the model satisifies the constraint by selecting a phrase containing an alternative noun or adjective:

BEFORE	…	*für*	***die***	***kommenden***	***ära***	*und*	*die*	…
		APPR	ART	ADJA	NN	KON	ART	
AFTER	…	*für*	***die***	***kommenden***	***epoche***	*und*	*die*	…
		APPR	ART	ADJA	NN	KON	ART	

In most of these cases, including this example, whilst the new phrase satisifies the constraint, it is actually incorrect. (Since both *Ära* and *Epoche* are singular feminine nouns, the adjective ending *-en* does not agree.) Examining the lexicon reveals that it contains a single, erroneous, plural entry for *epoche* (compared with 71 singular feminine entries).

In principle, a change of adjective or noun is not too worrying if the model considers it (or rather the encompassing phrase) a close-scoring alternative. A change of determiner is more likely to involve a semantic change and less likely to be acceptable. Fortunately, we only encounter one example:

BEFORE	”	*sie*	*war*	***meine***	***regulärer***	***abgeordneter***	.
	$PAR	PPER	VAFIN	PPOSAT	ADJA	NN	$.
AFTER	”	*sie*	*war*	***ein***	***regulärer***	***abgeordneter***	.
	$PAR	PPER	VAFIN	ART	ADJA	NN	$.

[2] This is clear from the English source text: ‘(in 2003, the same basket came to 6,800 forints.)’

Word Deletion

In a few cases, a constraint is satisifed by dropping a determiner:

BEFORE	*mit*	***dem***	***eindrucksvollen***	***sätzen***	,	*kommissionen*	,	...
	APPR	ART	ADJA	NN	$,	NN	$,	
AFTER	*mit*		***eindrucksvollen***	***sätzen***	,	*kommissionen*	,	...
	APPR		ADJA	NN	$,	NN	$,	

Constraint Circumvention

The decoder is more ingenious at satisfying constraints than we had anticipated. In the following example, the surface form words are unchanged, but an alternative part of speech choice is found for *die*, rendering the original rule non-applicable and allowing a weaker rule (ADJA NN in this case) to be used.

BEFORE	*aber*	***die***	***mikrobiologische***	***proben***	*von*	*patienten*	...
	KON	ART	ADJA	NN	APPR	NN	
AFTER	*aber*	***die***	***mikrobiologische***	***proben***	*von*	*patienten*	...
	KON	PRELS	ADJA	NN	APPR	NN	

Had we used a part-of-speech n-gram model, we would probably see fewer of these changes.

Non-Localised Changes

Generally, we do not want the application of our agreement constraint to affect the translation of words outside the noun phrase. All of the examples shown have been localised to the noun phrase, but in a significant minority this is not the case. Non-localised changes are mostly limited to one or two neighbouring words often involving the introduction or removal of words. Other times, the change is much wider-reaching.

Like for the changes to adjective and noun choice, we hope that in most cases, forcing an alternative phrase selection will, probabilistically, have little impact.

Classification of Change Types

As the examples above suggest, most of the changes brought about to satisfy our feature function are simple enough to interpret. Within the 1,026 sentence translations, 134 noun phrases are changed from the baseline. Table 5.2 shows a broad classification of the resulting changes and their frequency distribution in this sample.

Localised	Declensional	38.1%
	Alternative *nn* choice	5.2%
	Alternative *adja* choice	3.0%
	Alternative determiner choice	0.7%
	Dropped determiner	3.0%
	Change of POS	10.4%
	Combination of above	9.7%
Non-localised		29.9%

Table 5.2: Types and frequencies of change required to satisfy the agreement constraint

5.6 Agreement as a Soft Constraint

Hoping that our feature function's zealousness could be curbed by the log-linear model, and the most harmful translation changes avoided, we also tried re-tuning our system's weights, including `weight-a`. This actually lead to a decrease in BLEU score (12.14 against 12.19 for the baseline's and 12.27 for the hard constraint's) and a similarly reduced NIST score (4.7762 against and 4.7824 and 4.7859).

Since the evaluation score differences are so small, and the number of changes so minimal, we don't draw any conclusions from this. (Had the scores looked more interesting we would have applied a statistical significance test, such as bootstrap resampling (Koehn, 2004).)

In case we had made a poor choice of scoring system (recall that we used a fixed penalty for each non-agreeing phrase), we did also try defining an arbitrary binomial probability distribution over the number of agreeing / non-agreeing phrases in a hypotheses, which produced similar results, as did a re-tuned system in which `weight-a` was fixed at 1 (as for the hard constraint) throughout.

Unfortunately, the results are difficult to interpret manually as the change in weights for the model's other components leads to very different-looking translations. We did run the standalone agreement checker over the output and found the number of agreement failures had halved from 4.41% in the baseline to 2.20%.

Chapter 6

Conclusions and Further Work

6.1 Conclusions

We have proposed and implemented a simple unification-based agreement checker for German noun phrases. The agreement checker's lexicon is extracted from the training data of the translation system to which it is applied and so does not suffer from problems of domain-specificity. However, the raw lexicon produces a high failure rate when used to check the fluent training data and misses a significant proportion of genuine failures in translation output. By applying a small number of broad, but language-specific, transformations we are able to reduce this failure rate to well below 1%. We find varying failure rates on translation output, at around 6% and 4% of recognised noun phrases for two test sets translated by Moses using a hierarchical phrase-based grammar learned from the News Commentary corpus. A similar system trained on the Europarl corpus produced agreement failure rates of around 4% on the first translation set.

The checker can straightforwardly be integrated as a feature function into a log-linear SMT system. This feature function is narrowly focussed on a specific linguistic problem and applied as a hard constraint it has the desirable property that the effect on translation can easily be interpreted and analysed. In our small test set, we found that approximately 70% of changes from the baseline are localised to the single-noun phrases in question and that almost 40% are purely declensional changes.

There is much scope for improving the quality of the lexicon through further processing and we expect that these results can easily be improved upon.

6.2 Directions for Future Work

There are many minor extensions and variations to this work that we would have liked to explore and include in this dissertation:

- As suggested in section 4.5.4, a statistical model for detecting and removing anomalous feature structures from the lexicon. We would hope to see an increase in failure detection and an improvement in the performance of our feature function, especially through the removal of erroneous noun gender and number values, a problem we initially overlooked but later saw lead to undesirable translation changes (section 5.5.1).

- We would like to devise a method to evaluate the impact of our feature function on translation quality. One simple approach may be to employ larger test sets and to evaluate only those sentences that change between the baseline and the test system (in our 1,026-sentence test set, only 125 sentences are actually changed). Another may be to directly measure the change in probability assigned by the decoder.

- We would like to add prepositional phrase rules to our grammar. In German many prepositions require a succeeding noun phrase to take a specific case. For example, *mit* ('with') always takes the dative case. Whilst, to a limited extent, a language model will already implicitly encourage this (through the greater presence of *mit* in the histories of words declined for the dative case, for instance), we would hope to do better by explicitly accounting for this behaviour. This rule is trivial to implement in our grammar: a small set of single-noun prepositional phrase rules can be derived from a parsed corpus, as we did for noun phrases, and we can extend our lexicon processor to augment the feature structures of the small closed-set of prepositions with appropriate case values (and add additional feature stuctures, where a preposition can take varying cases).

- Our checker used tagged input to determine the rules. We would like to try using a purely surface-form model and searching for rule matches using the part-of-speech tags contained in the lexicon. We hope that the level of incorrect rule application would be minimal and would like to remove the dependency on tagged training data. (Though in German, an obvious fly in this ointment is the use of the same surface form words for the relative pronouns and definite articles.)

- Though we don't make any prediction either way, we would like to try the same approach in a non-hierarchical phrase-based model.

In the longer term, we would like to try applying a similar approach to a translation model that produces syntax. Whilst we have had reasonable success modelling German intra-noun phrase agreement with a trivial flat rule set, most agreement issues are less simple and testing agreement would be likely to require a richer syntactic context. Alternatively, a more powerful grammar could be developed for the checker, though parsing with a non-trivial rule set may become prohibitively expensive for use in a feature function.

Similarly, we anticipate that our exhaustive agreement search will not scale to longer-ranging agreement issues. We would like to explore the existing probabilistic approaches to unification-based grammars and parsing.

Bibliography

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990), *A Statistical Approach to Machine Translation* In Computational Linguistics 16(2).

Brown, P. F., Della Pietra, S. A., Della Pietra, V., and Mercer, R. L. (1993), *The Mathematics of Statistical Machine Translation: Parameter Estimation* In Computational Linguistics 19(2):263–311.

Brants, T., Popat, A., Xu, P., Och, F., and Dean, J. (2007), *Large Language Models in Machine Translation* In proceedings of EMNLP-CoNLL 2007, pp 858-867.

Chiang, D. (2005), *A Hierarchichal Phrase-Based Model for Statistical Machine Translation* In proceedings of ACL 2005, pp 263–270.

Collins, M., Koehn, P., and Kučerová, I. (2005), *Clause Restructuring for Statistical Machine Translation* In proceedings of ACL 2005.

Cowan, B., Kučerová, I, and Collins, M. (2006), *A Discriminative Model for Tree-to-Tree Translation* In proceedings of EMNLP 2006.

Durrell, M. (2002), *Hammer's German Grammar and Usage (Fourth Edition)* Arnold.

Grune, D. and Jacobs, C. J. H. (2008), *Parsing Techniques: A Practical Guide (Second Edition)*. Springer.

Johnson, S. (2008), *Exploring the German Language* Arnold.

Jurafsky, D. and Martin, J. H. (2008), *Speech and Language Processing (Second Edition)*. Pearson.

Kay, M. (1984), *Functional Unification Grammar: A Formalism for Machine Translation* In proceedings of Coling 1984.

Knight, K. (1997), *Automating Knowledge Acquisition for Machine Translation* In AI Magazine 18(4), 1997.

Koehn, P. (2004), *Statistical Significance Tests for Machine Translation Evaluation* In EMNLP 2004.

Koehn, P. (2005), *Europarl: A Parallel Corpus for Statistical Machine Translation* In MT Summit 2005.

Koehn, P. and Hoang, H. (2007), *Factored Translation Models* In EMNLP 2007.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., French, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007), *Moses: Open Source Toolkit for Statistical Machine Translation* Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June 2007.

Koehn, P., Bertoldi, N., Bojar, O., Callison-Burch, C., Constantin, A., Cowan, B., Dyer, C., Federico, M., Herbst, E., Hoang, H., Moran, C., Shen, W., and Zens, R. (2007), *Open Source Toolkit for Statistical Machine Translation: Factored Translation Models and Confusion Network Decoding.* Final Report of the 2006 Language Engineering Workshop, Johns Hopkins University.

Koehn, P. (forthcoming), *Statistical Machine Translation* Cambridge University Press.

Minkov, E., Toutanova, K., and Suzuki, H. (2007), *Generating Complex Morphology for Machine Translation* In proceedings of ACL 2007.

Manning, C. and Schütze, H. (1999), *Foundations of Statistical Natural Language Processing* The MIT Press, Cambridge, Massachusetts.

Och, F. J. (2003), *Minimum Error Rate Training in Statistical Machine Translation* In proceedings of ACL 2003.

Och, F. J. and Ney, H. (2002), *Discriminative Training and Maximum Entropy Models for Statistical Machine Translation* In proceedings of the 40th Annual Meeting for the Association for Comutational Linguistics (ACL).

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., and Radev, D. (2004), *A Smorgasbord of Features for Statistical Machine Translation* In Proceedings of the Joint Conference on Human Language Technology and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL).

Shieber, S. (1986), *An Introduction to Unification-Based Approaches to Grammar*, Volume 4 of CSLI Lecture Notes Series. Center for the Study of Language and Information, Stanford, CA.